Санкт–Петербургский государственный университет

Холодаева Екатерина Владимировна

Выпускная квалификационная работа

Прогнозирование рисков кардиальных патологий на основе массивов данных результатов взятых анализов (проектная работа)

Уровень образования: бакалавриат
Направление 02.03.03 «Математическое обеспечение и администрирование информационных систем»
Основная образовательная программа СВ.5006.2017 «Математическое обеспечение и администрирование информационных систем»

Научный руководитель:
доцент кафедры информатики,
кандидат психологических наук
Татьяна Валентиновна Тулупьева
Рецензент:
доцент факультета информационных
технологий и программирования университета ИТМО,
кандидат технических наук
Иван Борисович Сметанников

Санкт-Петербург
2021 г.

Saint Petersburg State University

Kholodaeva Ekaterina Vladimirovna

Graduate qualification work

Predicting the risks of cardiac pathologies based on the dataset of test results (project work)

Education level: bachelor's degree
Direction 02.03.03 «Software and Administration of Information Systems Software Engineering»
Basic educational program CB.5006.2017 «Software and Administration of Information Systems Software Engineering»

Scientific supervisor:
Associate Professor, Computer Science Chair
Candidate of Psychology,
Tatiana Valentinovna Tulupyeva
Scientific adviser:
Associate Professor,
Faculty of Information Technology and Programming,
ITMO University,
PhD in Engineering Sciences,
Ivan Borisovich Smetannikov

Saint-Petersburg
2021

Содержание

Введение

Актуальность темы. Ежегодно от сердечно-сосудистых заболеваний(ССЗ) умирает больше людей, чем от любой другой болезни, что подтверждается информацией, приведенной на сайте Всемирной организации здравоохранения(ВОЗ) [24]. По оценкам ВОЗ, в 2016 году от ССЗ умерло 17,9 миллиона человек, что составило 31% всех случаев смерти в мире. 85% этих смертей произошло в результате сердечного приступа и инсульта [1].

Данная работа основана на информации, полученной от «Всероссийского центра экстренной и радиационной медицины им. А.М.Никифорова» МЧС России (ВЦЭРМ). При поступлении у некоторых пациентов были обнаружены симптомы ССЗ. Впоследствии у них произошло резкое ухудшение состояния здоровья. И у части из них эти осложнения привели к гибели. Требуется заблаговременно определять такие случаи, чтобы у докторов была возможность предотвратить смерти людей. Результаты данной работы позволят врачам понять, каким пациентам стоит уделять больше внимания, а также выделить статистически значимые признаки, которые имеют наибольшее значение при постановке диагноза пациенту.

Целью данного проекта является автоматизация оценки риска смерти пациента с симптомами ССЗ на основе его личных данных и данных медицинских исследований с применением методов машинного обучения.

Для выполнения обозначенной цели были выделены следующие задачи:

1. исследовать существующие методы машинного обучения и их использование в сфере сердечно-сосудистых заболеваний;

2. реализовать несколько моделей предсказания, произвести их сравнительный анализ;

3. разработать критерий с целью использования показателей ЭКГ для обучения моделей;

4. разработать архитектуру прототипа программного модуля для визуализации оценки предсказания моделей и реализовать полученные методы в данном прототипе.

Объектом исследования являются личные данные пациентов и сведения об изменении их сердечного ритма во время наблюдения, полученные от ВЦЭРМ.

Предметом исследования являются алгоритмы и методы предсказания смерти человека на основе этих данных.

Научная новизна. Все результаты, выносимые на защиту, являются новыми. Впервые был получен программный комплекс на Python, позволяющий по вводимым пользователем медицинским показателям пациента, поступившего с симптомами ССЗ, получить вероятностную оценку итога ухудшения его состояния.

Теоретическая и практическая значимость исследования. Реализация приложения на языке Python позволит врачам получить вероятность смерти человека, и, как следствие, уделить внимание пациентам, чья жизнь находится под угрозой. А также обратить внимание на признаки, которые являются наиболее значимыми при постановке диагноза.

Методология работы заключается в разработке архитектуры и реализации программы, описании реализуемых методов и алгоритмов, создании графического интерфейса для получения данных о пациенте, их обработке с последующей интерпретацией и выведением результата.

Методы. Для программной реализации приложения были использованы методы объектно-ориентированного программирования. Программная реализация осуществлялась в среде разработки PyCharm на языке программирования Python с использованием фреймворка PyQT для визуализации.

Структура и объём работы. Текст работы включает в себя введение, четыре главы, заключение и список литературы. Общий объём ВКР — 41 страница.

В главе 1 обосновывается актуальность задачи и приводится описание целей и задач данной работы

В главе 2 приведен обзор существующих решений похожих задач в данной области, а также представлено описание алгоритмов, используемых для поставленной задачи.

В главе 3 представлены результаты работы алгоритмов, которые бы-

ли реализованы для решения поставленной задачи, а также сравнительный анализ моделей.

Глава 4 посвящена программной реализации моделей предсказания, а также описанию программного комплекса.

Глава 1. Описание предметной области

1.1 Введение

Сердечно-сосудистые заболевания (ССЗ) — это группа болезней сердца и кровеносных сосудов. И именно от них в настоящее время умирает людей больше, чем от любой другой болезни [24].

Каждому пациенту с симптомами ССЗ назначают электрокардиографию (ЭКГ) для диагностики заболевания. Это очень важное исследование, так как позволяет врачам получить информацию о состоянии сердца человека и диагностировать многие ССЗ (ишемия, стенокардия, аритмия). ЭКГ представляет собой бесконтактный процесс записи электрических импульсов сердца в течение определенного периода времени с использованием электродов, размещенных на поверхности кожи [26]. Каждое сердечное сокращение представляет собой определенный рисунок. На графике 1 представлена схема сердечного ритма.

Информация, предоставленная ВЦЭРМ содержит данные о возрасте, поле, днях госпитализации пациентов, сведения о ритме, а также дневники ЭКГ (1 дневник – 1 исследование), в каждом из которых от 1 до 17 измерений интервалов RR и QT.

RR интервал – интервал между сердечными сокращениями человека. Чаще всего он используется для оценки вариабельности сердечного ритма.

QT интервал – часть цикла ЭКГ, которая отражает время, необходимое для деполяризации и реполяризации миокарда желудочков [26]. Это один из наиболее значимых показателей, так как увеличение или уменьшение данного показателя может говорить о возникновении опасных для жизни аритмий.

У пациентов с симптомами ССЗ, поступивших в центр экстренной

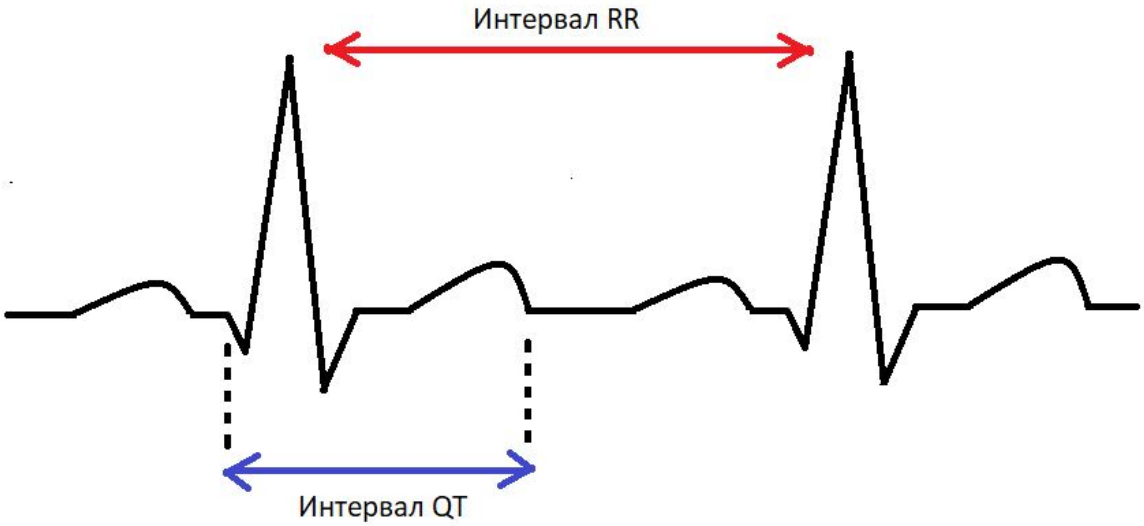

Рис. 1: Схема сердцебиения человека

и радиационной медицины, спустя определенное время произошло резкое ухудшение состояния здоровья. Часть из них выжили, у остальных – осложнения привели к гибели. Хочется заблаговременно определять такие случаи. Но при поступлении пациента с симптомами ССЗ сложно с первого взгляда определить, какие из показателей человека играют более важную роль при постановке диагноза, и кто из пациентов находится в большей опасности. Своевременное выявление предикторов возможной смерти пациентов поможет уменьшить количество летальных исходов, а визуализация результатов позволит значительно облегчить врачам работу с обследуемыми людьми.

1.2 Методы машинного обучения

Машинное обучение – мощный инструмент анализа данных. Оно широко используется в медицине для ряда задач: классификация результатов МРТ для диагностики опухолей, генерация молекулярных структур (генерация молекул, которые потенциально могут быть лекарствами) и для многих других. Но на данный момент не существует решения для задачи предсказания смерти пациента уже имеющего симптомы ССЗ и без кри-

тических показателей (пожилой возраст, сахарный диабет) на основе его личных данных и данных ЭКГ. Это поможет врачам понять, кому из пациентов необходимо уделить повышенное внимание и какие из показателей играют большую роль в постановке диагноза.

В данной работе для решения поставленной задачи используются такие методы машинного обучения, как логистическая регрессия, метод опорных векторов, метод решающего дерева и метод случайного леса.

1.3 Цели и задачи

Описываемые выше данные позволяют сделать вывод о необходимости реализации алгоритма предсказания ухудшения состояния пациента с симптомами ССЗ на основе личных данных(пол, возраст, ритм сердца) и данных медицинских исследований, а также визуализации результата для удобства использования врачами.

Целью данного проекта является автоматизация оценки риска смерти пациента с симптомами ССЗ на основе его личных данных и данных медицинских исследований с применением методов машинного обучения.

Для выполнения обозначенной цели были выделены следующие задачи:

1. исследовать существующие методы машинного обучения и их использование в сфере сердечно-сосудистых заболеваний;

2. реализовать несколько моделей предсказания, произвести их сравнительный анализ;

3. разработать критерий с целью использования показателей ЭКГ для обучения моделей;

4. разработать архитектуру прототипа программного модуля для визуализации оценки предсказания моделей и реализовать полученные методы в данном прототипе.

Глава 2. Используемые подходы и решения

В данной главе описан анализ работ по схожей тематике, а также представлено описание основы исследования.

2.1 Релевантные работы

Машинное обучение широко применяется в медицинской сфере и в области ССЗ в частности. Например, распознавание опухолей или исследование факторов риска появления ССЗ. В статьях [5, 6], описаны методы машинного обучения для диагностики легочной гипертензии и прогнозирования сердечно-сосудистых заболеваний. Прогнозирование сердечно-сосудистых заболеваний на основе данных из электронных медицинских карт представлено в статье [7].

В статье [8] описывается исследование, направленное в основном на разработку системы, которая сможет обнаруживать сердечно-сосудистые заболевания у пациента, используя подходы машинного обучения на основе возраста, наличия боли в груди, электрокардиограммы, давления и других данных пациентов. Авторы сравнили результаты работы методов глубокого машинного обучения.

Цель статьи [9] заключалась в нахождении нового метода прогнозирования сердечно-сосудистых заболеваний, основанный на характеристиках ЭКГ и ФКГ. Описание таких моделей машинного обучения как решающее дерево, случайный лес и некоторых других приводится в статье [10]. Они используются авторами в рамках задачи обнаружения и классификации стеноза аорты с использованием частотно-временных характеристик кардиомеханических сигналов грудной клетки, полученных от носимых датчиков. В [11] авторами поднимается проблема большого количества смертей, причинами которых стали сердечно сосудистые заболевания, в частности, ишемическая болезнь. Авторы предлагают решение этой проблемы с помощью методов машинного обучения. В статье [12] исследуется возможность применения метода дерева Хёффдинга и метода случайного леса для возможности предсказания сердечно-сосудистых заболеваний путем анализа различных медицинских параметров, таких как частота сердечных сокра-

щений, уровень холестерина, HbA1c, вес, результаты ЭКГ.

В [13] авторы в рамках задачи прогнозирования сердечно-сосудистых заболеваний на основе биологических образцов провели сравнение моделей метода опорных векторов, дерева решений, нейронной сети и K-ближайших соседей.

Однако во всех вышеописанных статьях исследование было нацелено только на возможное появление ССЗ в будущем, пациенты с симптомами заболеваний были исключены из исследования, и для предсказания использовались результаты анализов, на получение которых требовалось длительное время.

Задача стратификации риска смертности описана в статье [14]. В данном исследовании проводится изучение основных предикторов смерти от ССЗ. Но пожилой возраст, сахарный диабет, которые являются главными предикторами, слишком сильно уменьшают диапазон обследуемых пациентов, поэтому необходимо получить предикторы, которые будут применимы ко всем пациентам, независимо от возраста и побочных заболеваний.

На основе анализа данных статей были выбраны методы, используемые в работе: логистическая регрессия, метод опорных векторов, дерево решений и метод случайного леса. Именно эти модели чаще всего встречались в релевантных работах.

В основу данной работы легли результаты, полученных в статье коллектива лаборатории ТиМПИ СПб ФИЦ РАН. В данном исследовании описывается критерий, полученный из сумм интервалов RR и QT пациентов.

2.2 Используемые теоретические методы

В данной работе для прогнозирования исхода ухудшения состояния пациента использовались следующие методы: логистическая регрессия, метод опорных векторов, метод решающего дерева и метод случайного леса. Для анализа результатов моделей прогнозирования использовались ROC-анализ, кроссвалидация и оценка точности модели классификации.

2.2.1 Логистическая регрессия

Предсказание происходит посредством его сравнения с логистической кривой [16]. В качестве ответа модель выдает 0 или 1 (произойдет событие или нет) [16]. Для прогнозирования вероятности появления события необходимо иметь набор признаков [16].

Описание модели

y - переменная, которая определяет, произошло прогнозируемое событие или нет. Она называется зависимой и может принимать, соответственно, два значения 0 и 1.

$x_1, ..x_n$ - набор переменных, которые определяются признаками, на основе которых вычисляется вероятность принятия зависимой переменной определенного значения. Эти переменные называются независимыми.

$f(x) = \frac{1}{(1+e^{-x})}$ - логистическая функция [16].

Теперь, сделаем предположение, что
$\mathbb{P}\{\, y = 1 \,|\, x \,\} = f(z)$, где
$f(z)$ – логистическая функция,
$z = \theta^T * x = \theta_0 + \theta_1 * x_1 + \ ... \ + \theta_n * x_n$
x – вектор значений независимых переменных $1, x_1, ..x_n$.
θ – вектор значений коэффициентов регрессии $(\theta_1, ..\theta_n)$.

И так как y принимает только два значения, знаем, что
$\mathbb{P}\{\, y = 0 \,|\, x \,\} = 1 - f(z) = 1 - f(\theta^T * x)$

Далее необходимо составить обучающую выборку для того, чтобы правильно подобрать коэффициенты регрессии $(\theta_1, ..\theta_n)$ [16]. И с помощью метода максимального правдоподобия (чаще всего используется именно он) выбираются параметры $(\theta_1, ..\theta_n)$, максимизирующие значение функции правдоподобия на обучающей выборке [16].

2.2.2 Метод опорных векторов

Основная идея метода — построить такую гиперплоскость в пространстве, которая разделит объекты, принадлежащие разным классам

наиболее оптимальным способом [17]. Чем больше расстояние от объектов до гиперплоскости, тем лучше работает модель [17]. Вектора, которые ближе остальных лежат к гиперплоскости, называются опорными векторами [17].

Описание модели

Обучающая выборка – множество пар, таких что:
$(x_1, y_1), ..., (x_n, y_n)$, $x_i \in \mathbb{R}^n$, $y_i \in \{-1, 1\}$ [17].

Так как любая гиперплоскость может быть задана в виде $< w, x > + b = 0$ для каких-то w и b, при использовании метода опорных векторов строится классифицирующая функция F в виде
$F(x) = sign(< w, x > + b)$, где
$<, >$ – скалярное произведение,
w – нормальный вектор к разделяющей гиперплоскости,
b – вспомогательный параметр

Объекты, для которых $F(X) = 1$ – попадают в 1 класс, а для которых $F(X) = -1$ – в другой [17].

Хотим выбрать такие w и b, которые максимизируют расстояние до каждого объекта. Данное расстояние равно $\frac{1}{\|w\|}$ (рисунок 2). Найти максимум для $\frac{1}{\|w\|}$ – это то же, что и найти минимум для $\|w^2\|$ [17].

Далее решаем задачу оптимизации: $\begin{cases} \arg\min_{w,b}\|w\|^2, \\ y_i(< w, x_i > + b) \geqslant 1,\ i = 1, ..., n. \end{cases}$

А это стандартная задача квадратичного программирования, решается с помощью множителей Лагранжа [17].

Теперь наша квалифицирующая функция : $F(x) = sign(< w, \phi(x) > + b)$,

Выражение $k(x, x') =< \phi(x), \phi(x') >$ называется ядром классификатора [17]. Я дром может быть любая положительно определенная функция двух переменных. Но она должна быть симметричной [17]. От выбора ядра зависит точность классификатора [17]. Возможные виды ядер:

1. линейное $k(x, x') =< x, x' >$;

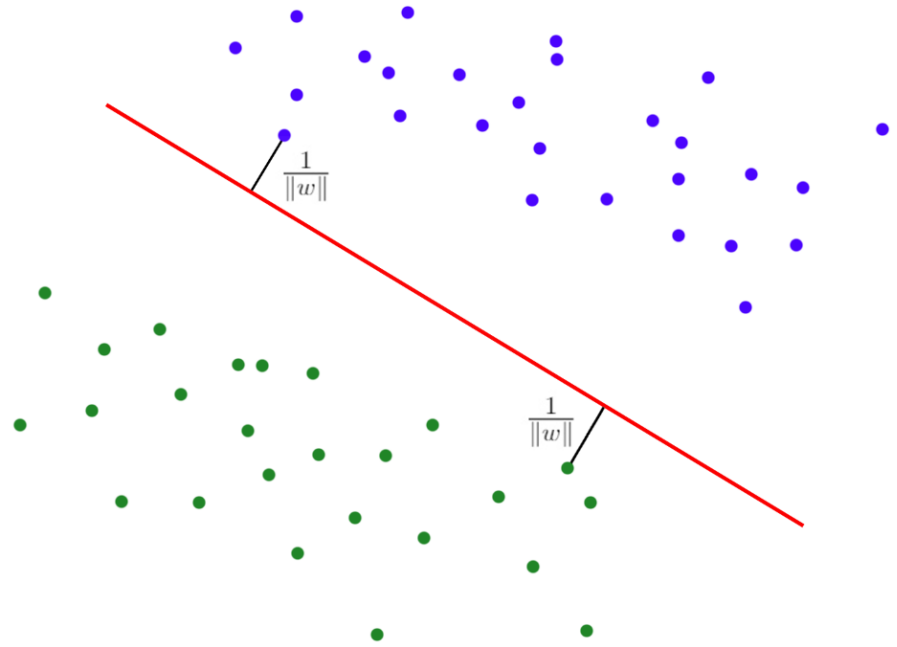

Рис. 2: Пример выбора гиперплоскости

2. радиальная базисная функция $k(x, x') = e^{-\phi\|x-x'\|^2}$, $\phi > 0$;
3. радиальная базисная функция Гаусса $k(x, x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$;
4. полиномиальное $k(x, x') = (< x, x' > + const)^d$;
5. сигмоид $k(x, x') = tanh(k < x, x' > +c)$, $k > 0$, $c < 0$;

2.2.3 Дерево решений

Дерево решений представляет собой древовидную структуру, которая состоит из правил "Если.., то.., иначе..."[16]. В процессе обучения правила генерируются за счет обобщения множества обучающих случаев [16].

Определения, необходимые для работы с деревом решений:

Решающее правило – некоторая функция от объекта, которая позволяет определить в сторону какой из дочерних вершин двигаться дальше. Имеет вид "Если.., то.."[16].

Объект – рассматриваемый случай [16].

Структура

Дерево решений – это способ представления решающих правил в определенной иерархии [16]. Дерево включает в себя узлы и листья (конечные узлы дерева) [16]. Узлы включают в себя решающие парвила и производят проверку объектов на соответствие определенного атрибута [16]. На рисунке 3 изображен пример решающего дерева.

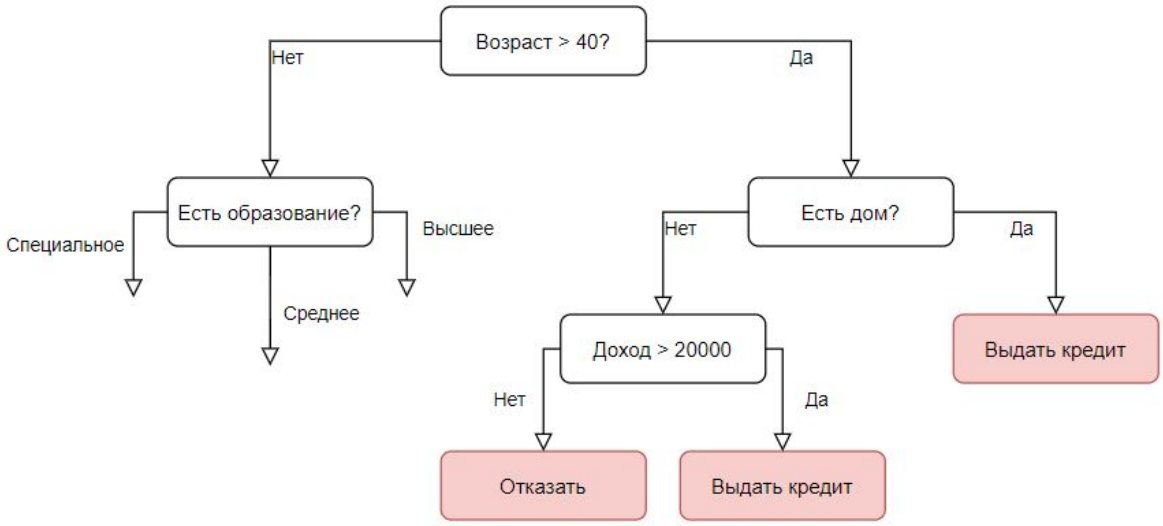

Рис. 3: Пример решающего дерева

Построение дерева

Главная задача при построении дерева – рекурсивно разбить обучающее множество на классы с применением узлов с решающими правилами [16].

Самые известные алгоритмы обучения:

1. ID3 (Iterative Dichotomizer 3). При использовании данного алгоритма используется энтропийный критерий. Дерево строится до тех пор, пока в каждом листе не окажутся объекты одного класса или же пока дальнейшее разбиение позволяет уменьшать энтропийный критерий [16].

2. C4.5. При использовании данного алгоритма используется Gain Ratio критерий. Дерево строится до тех пор, пока количество объектов в

листе не достигнет заданного ограничения.Также позволяет работать с пропущенными значениями атрибутов [16].

3. CART (Classification and Regression Tree). При использовании данного алгоритма используется критерий Джини [16]. Строятся деревья, каждый узел которого может иметь только 2 потомка [16].

2.2.4 Случайный лес

Случайный лес – алгоритм машинного обучения, заключающийся в использовании множества решающих деревьев [17]. Основная идея заключается в том, что каждое из деревьев само по себе может давать не очень высокую точность классификации, но за счет их большого количества результат модели оказывается достаточно хорошим [17].

Алгоритм обучения классификатора

Пусть обучающая выборка состоит из n пар (x, y), где x – вектор признаков размера m, y – показывает принадлежность к определенному классу и принимает значение 0 или 1. Зададим параметр l. Обычно $l = \sqrt{m}$.

Для обучения каждого дерева леса выделяют следующие шаги:

1. Из обучающей выборки выбираются n элементов с повторениями. Некоторые элементы могут попасть туда несколько раз, а некоторые не попадут в нее совсем (такие элементы называются out-of-bag) [17].

2. Строится решающее дерево, классифицирующее элементы полученной выборки. В процессе создания узла дерева случайным образом выбирается набор атрибутов размером l, в соответствии с которыми производится разбиение. Выбор оптимального из l атрибутов может производится с помощью любого критерия [17].

3. Дерево строится, не используя отсечение ветвей до конца выборки [17].

Объекты классифицируются путем голосования [17]. Каждое построенное дерево определяет объект к одному из классов [17]. Тот класс, за который проголосовало большее количество деревьев выигрывает [17].

2.3 Кроссвалидация

Кроссвалидация – это проверка того, насколько модель предсказания применима при работе с независимым набором данных. Она используется для сравнения алгоритмов и их оценки путем разделения исходных данных на несколько частей. Некоторые части используются для обучения модели, а другие для проверки [28].

В процессе кроссвалидации происходит подбор гиперпараметров. Гиперпараметры – это параметры, которые не могут быть настроены во время обучения модели. Например, для логистической регрессии гиперпараметром будет максимальное число итераций.

Тренировочная часть данных делится на n частей. Далее, мы обучаем модель на каждых $n-1$ частях с определенным набором гиперпараметров и тестируем на оставшейся. Затем усредняем результат. Таким образом мы находим лучшую комбинацию гиперпараметров, сравнивая получившиеся средние значения. Это занимает больше времени, но позволяет повысить качество классификации.

2.4 Метрики

Чтобы оценить количество ошибок первого и второго рода, опасных для медицины была построена матрица неточностей для каждой модели [27].

2.4.1 Матрица неточностей

Матрица неточностей – это матрица $N \times N$. Она используется для оценки модели и и позволяет оценить качество работы модели и то, какие ошибки она допускает. На рисунке 4 приведена матрица для задачи двоичной классификации.

TN (true-negative rate) – истинно отрицательные случаи (верно классифицированные отрицательные случаи).

FN (false-negative rate) – ложно отрицательные случаи (положительные случаи, классифицированные как отрицательные) – ошибка первого рода.

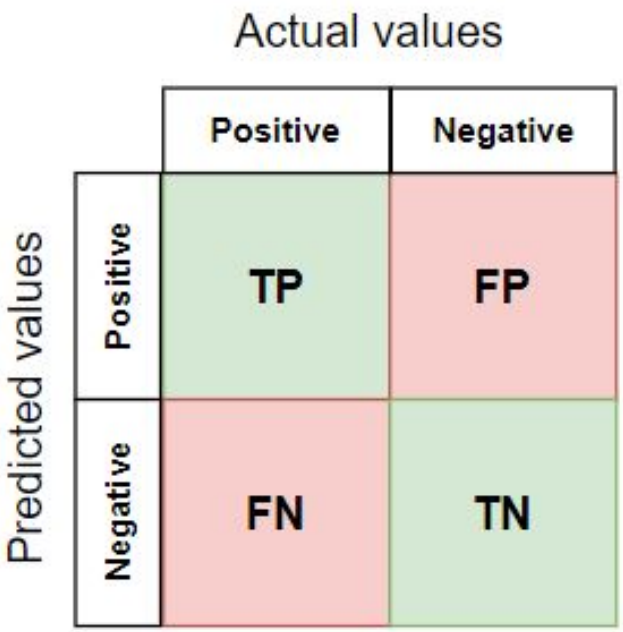

Рис. 4: матрица для задачи двоичной классификации

TP (true-positive rate) – истинно положительные случаи (верно классифицированные положительные случаи).

FP (false-positive rate) – ложно положительные случаи (отрицательные случаи, классифицированные как положительные) – ошибка второго рода.

Самые опасные в области медицины ошибки первого рода, так как пациенту, чье здоровье в опасности, не уделяют достаточно внимания.

2.4.2 Precision, Recall, F-мера

Precision – метрика, определяющая сколько случаев, которые были классифицированы как положительные, действительно являются положительными.

$Precision = \frac{TP}{TP+FP}$

Recall – метрика, определяющая сколько положительных случаев из всех объектов, определенных как положительные, модель смогла правильно определить.

$Recall = \frac{TP}{TP+FN}$

F-мера – совместная оценка precision и recall.
$F - score = 2 * \frac{Precision * Recall}{Precision + Recall}$

2.4.3 ROC-анализ

ROC (Receiver Operator Characteristic) — кривая, которая применяется для представления результатов бинарной классификации. Она отображает зависимость между TP и FP.
Качество классификации оценивается с помощью AUC (area under the curve) – площадь под ROC-кривой. Существует шкала значений AUC, по которой можно судить о качестве классификации (таблица 1).

Значение AUC	Качество классификации
0.9-1.0	Отличное
0.8-0.9	Очень хорошее
0.7-0.8	Хорошее
0.6-0.7	Среднее
0.5-0.6	Неудовлетворительное

Таблица 1: Шкала значение AUC

Глава 3. Теоретическая часть

В этой главе представлены основные теоретические результаты выпускной работы бакалавра. Она посвящена описанию данных, на которых проводилось тестирование методов, обозначенных в главе 2 и представлены результаты применения методов к задаче предсказания итога ухудшения здоровья пациента.

3.1 Описание данных

Для обучения моделей использовались данные, предоставленные ВЦЭРМ [18]. Данные представляют собой 2 таблицы, первая содержит такие характеристики пациента как пол, возраст, критерий, сведения об изменении ритма (таблица 2).

Номер пациента	Группа пациента	Пол	Возраст
$i : i \in 1..151$	$g : g \in \{0, 1\}$, 0 – пациент умер, 1 – пациент выжил	$p : p \in \{0, 1\}$, 0 – женщина, 1 – мужчина	$a : a \in 34..94$

Было ли изменение ритма?	Вид изменения ритма
$c : c \in \{0, 1\}$ 0 – ритм менялся, 1 – ритм не менялся	$v : v \in \{1, 2, 3\}$ 1 – синусовый ритм 2 – фибрилляция предсердий 3 – ритм менялся

Таблица 2: Общий вид таблицы, содержащей информацию о пациентах.

Модели машинного обучения обычно принимают на вход вектор фиксированной размерности. Но каждому пациенту проводили несколько исследований ЭКГ, поэтому количество интервалов RR и QT разное. Чтобы обеспечить соответствие каждому пациенту только одного вектора признаков одинаковой размерности в рамках предыдущих исследований Командой исследователей лаборатории теоретических и междисциплинарных проблем информатики Санкт-Петербургского института информатики и автоматизации РАН (ТиМПИ СПб ФИЦ РАН) был разработан критерий,

который отражает зависимость группы пациента от сумм интервалов RR и QT и рассчитывается по формуле:

$$\log_{\sum_{i=1}^{n} QT^3} \sum_{i=1}^{n} RR^3,$$

где n - количество измерений.

Вторая таблица содержит информацию об интервалах RR и QT (таблица 3).

Номер пациента	Интервал RR	Интервал QT
$i : i \in 1..120$	$rr : rr \in 300..1300$	$qt : qt \in 250..810$

Таблица 3: Общий вид таблицы, содержащей информацию о пациентах.

Описательная (первичная) статистика по полученным данным:

- всего пациентов в таблице 2 – 151;
- пациентов, принадлежащих группе 1 (умершие) – 76 (49,7% от общего количества);
- пациентов, принадлежащих группе 0 (выжившие)– 77 (51,3% от общего количества);
- для данных пациентов проводилось ЭКГ – 670 раз (от 1 до 38 раз у одного человека);
- распределение мужчин к женщинам (в процентах) – 48,3% и 51.7%;
- на графике 5 изображено распределение по возрасту;
- на диаграмме 6 изображено распределение диагнозов

При составлении датасета для балансировки данных каждому пациенту из группы 0 сопоставлялся представитель группы 1 близкий по симптоматике, возрасту и дате поступления.

Размер данных, предоставленных ВЦЭРМ, очень маленький. Это естественно для медицинских данных, так как размер датасета напрямую зависит от количества умерших пациентов и от количества людей с определенными симптомами. Поэтому, чтобы избежать переобучения модели, были использованы следующие методы [25]:

Рис. 5: Распределение диагнозов

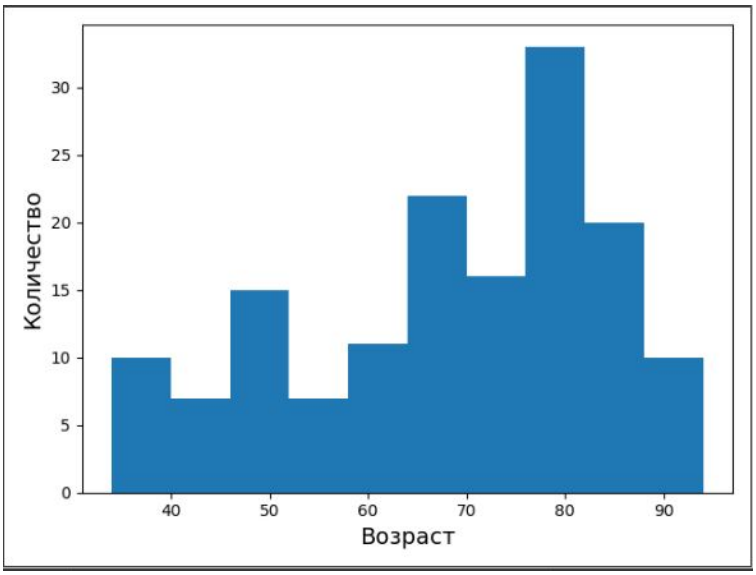

Рис. 6: Распределение пациентов по возрасту

- Использование простых моделей и моделей с небольшим числом параметров. Также для древовидных моделей уменьшение их максимальной глубины. Это позволит ограничить способность модели видеть закономерности и несуществующие связи.

- Уменьшение количества выбросов. Выбросы – значения, которые далеко выходят за рамки других наблюдений.
- Удаление параметров, которые не влияют на прогнозирование.
- Комбинирование моделей.

3.2 Подготовка данных

Чтобы избежать неверной классификации, перед применением к полученным данным методов классификации необходимо их подготовить.

Из датасета был удален столбец "Номер пациента", так как он не влияет на результат работы модели.

Также из датасета были удалены пациенты, у которых значения некоторых признаков можно считать выбросом (например, человек, чей возраст равен 0).

Далее данные были разделены на 5 частей. 4 части использовались как обучающая выборка и 1 часть как тестовая.

При разделении датасета используется параметр random_state. Он управляет перемешиванием, применяемым к данным. То есть изменение этого показателя приведет к различным разделениям на обучающую и тестовую выборки.

Чтобы избежать случайно получившегося хорошего или плохого результатов модели, которые зависели бы от конкретных выборок, был организован цикл от 0 до 100 по параметру random_state, и точность подсчитывалась для каждой из полученных моделей.

Средняя точность модели подсчитывалась как среднее арифметическое всех полученных точностей.

3.3 Результаты применения моделей

Для каждой модели экспериментальным путем были выбраны оптимальные параметры – это параметры, при которых модель показала лучшие результаты при кроссвалидации. Для этого были использованы программные средства библиотеки scikit-learn.

3.3.1 Логистическая регрессия

В рамках данной задачи зависимая переменная y определяет попадет пациент в основную или же в контрольную группу. Независимыми параметрами $x_1, ..x_n$ были такие характеристики пациента, описанные в датасете. Рассчитанные коэффициенты регрессии приведены в таблице 4

Параметр	Коэффициент регрессии
Возраст	0.00405717
Пол	0.07427747
Критерий	0.17076263
Было ли изменение ритма	-0.93851999
Ритм	0.03648514

Таблица 4: Таблица коэффициентов логистической регрессии

Оптимальными параметрами для данной модели являются:
максимальное число итераций – 100;
алгоритм решения задачи оптимизации – liblinear;
среднее значение точности предсказаний модели - 0.6265;
лучшее значение точности предсказаний модели - 0.8065.

На рисунке 7 изображена ROC-кривая. Значение AUC – 0.7991452991452992. Качество модели – хорошее.

3.3.2 Метод опорных векторов

В рамках данной задачи классами являются основная и контрольная группа.
Оптимальными параметрами для данной модели являются:
Ядро – линейное.
Среднее значение точности предсказаний модели - 0.6161
Лучшее значение точности предсказаний модели - 0.7742

На рисунке 8 изображена ROC-кривая. Значение AUC – 0.8136363636363637. Качество модели – очень хорошее.

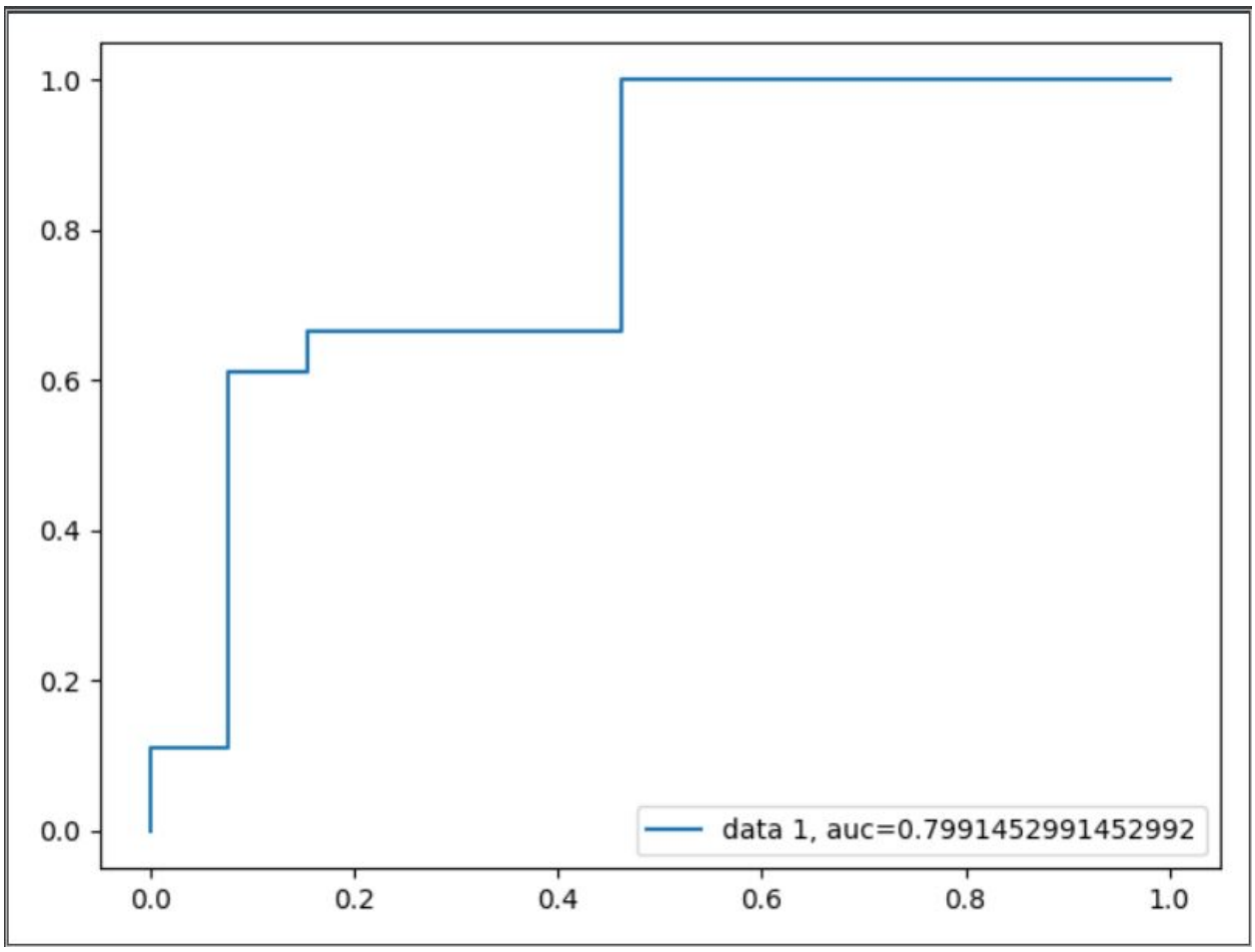

Рис. 7: ROC-кривая для логистической регрессии

3.3.3 Дерево решений

Оптимальными параметрами для данной модели являются:
критерий выбора атрибута для разбиения – Джини;
максимальная глубина дерева – 4;
ограничение на число объектов в листьях – 6.
Среднее значение точности предсказаний модели - 0.6467483870967743;
Лучшее значение точности предсказаний модели - 0.8065.

На рисунке 9 изображена ROC-кривая. Значение AUC – 0.8375. Качество модели – очень хорошее.

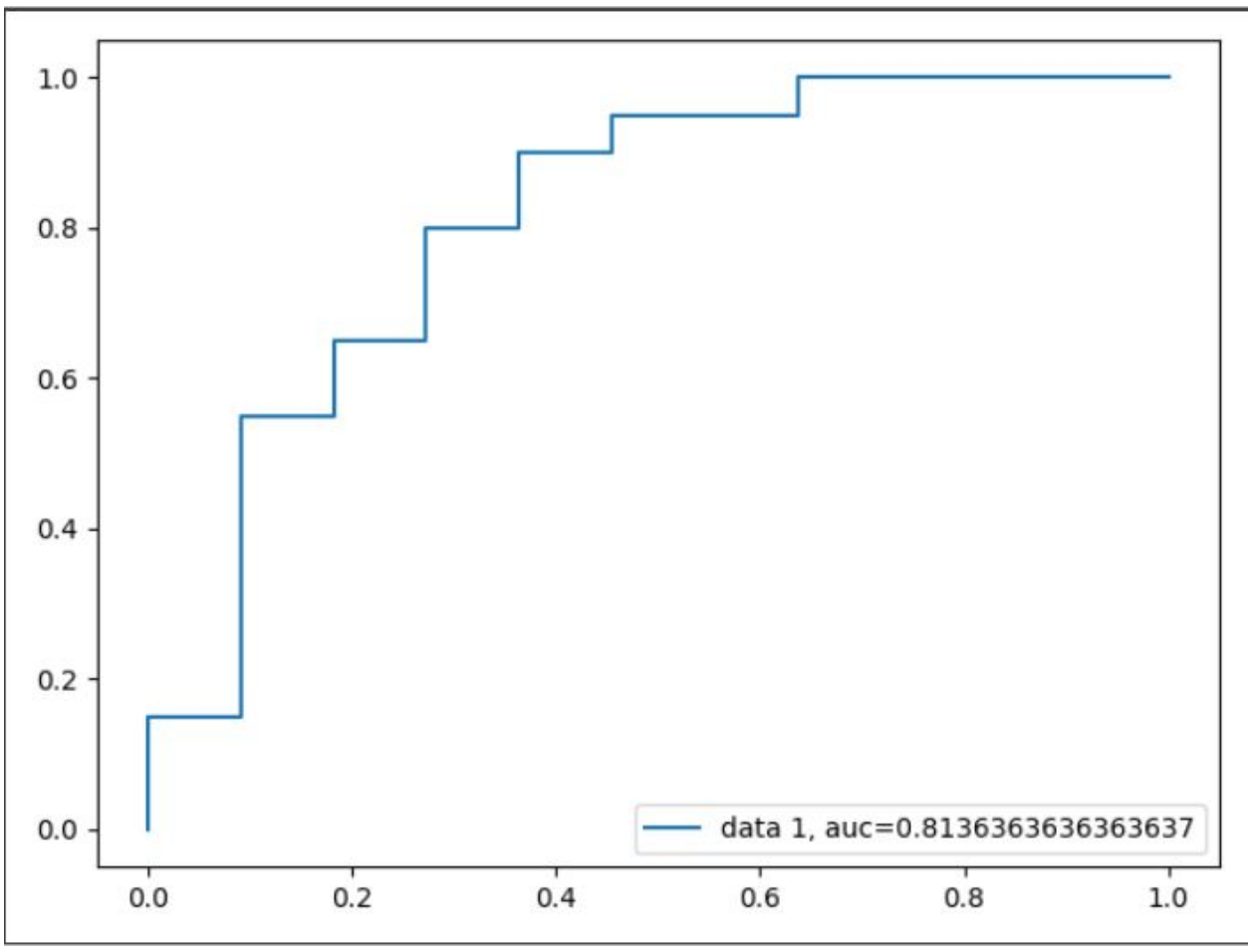

Рис. 8: ROC-кривая для метода опорных векторов

3.3.4 Случайный лес

Оптимальными параметрами для данной модели являются:
критерий выбора атрибута для разбиения – энтропийный;
максимальная глубина дерева – 11;
ограничение на число объектов в листьях – 1;
количество деревьев – 103.
Среднее значение точности предсказаний модели - 0.678322580645161.
Лучшее значение точности предсказаний модели - 0.8387096774193549.

На рисунке 10 изображена ROC-кривая. Значение AUC – 0.9125. Качество модели – очень хорошее.

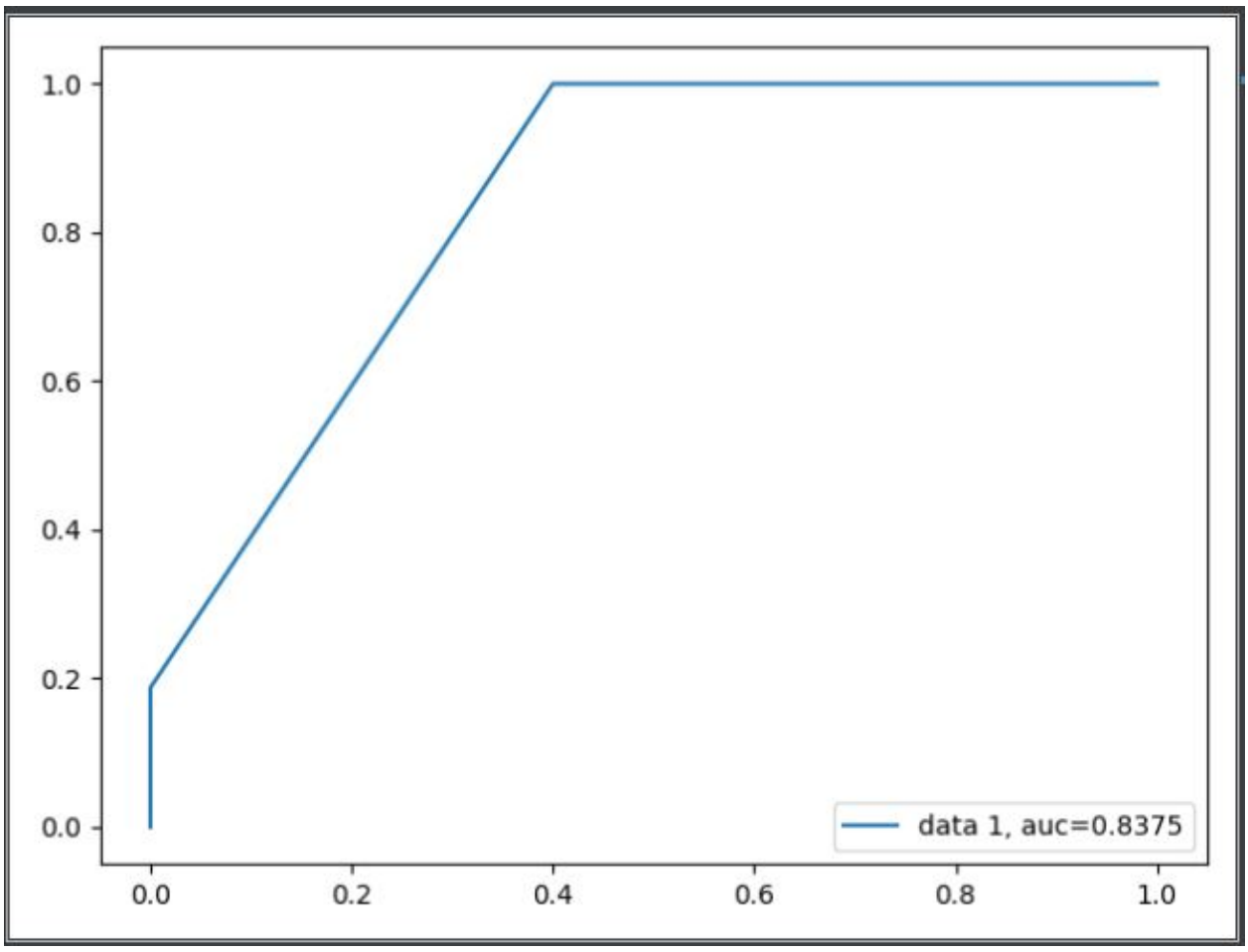

Рис. 9: ROC-кривая для метода решающих деревьев

3.4 Анализ результатов

В рамках данной работы были реализованы 4 предсказательные модели:

1. логистическая регрессия;
2. метод опорных векторов;
3. дерево решений;
4. метод случайного леса.

В таблице 5. приведены средние точности вышеописанных моделей. Анализ результатов применения моделей позволяет сделать вывод, что наилучший результат показала модель случайного леса.

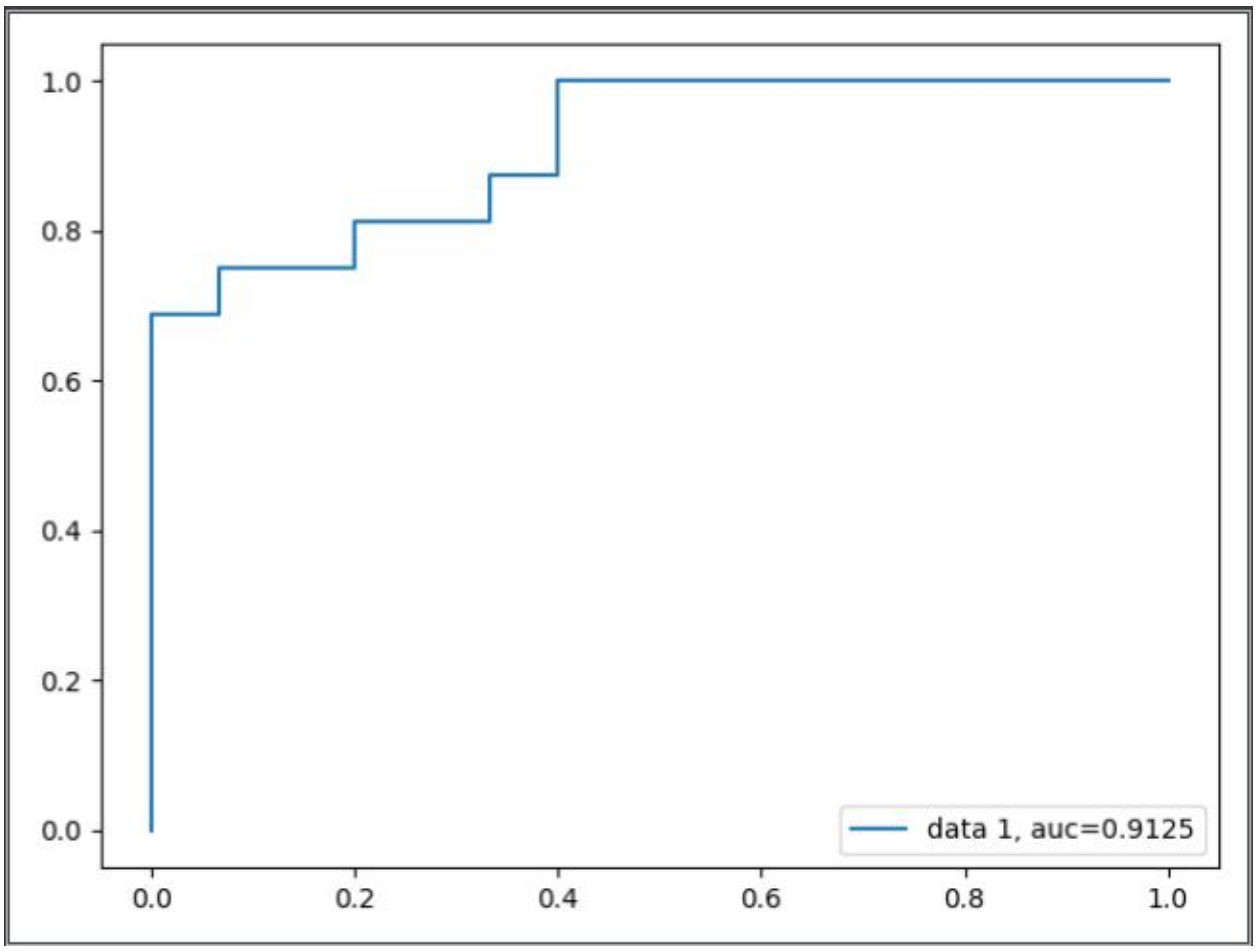

Рис. 10: ROC-кривая для метода случайного леса

Модель	Логистическая регрессия	Метод опорных векторов	Дерево решений	Метод случайного леса
Точность	0.6265	0.6161	0.6467	0.6783

Таблица 5: Точности предсказательных моделей.

3.5 Метрики

Значения матрицы неточностей для каждой из моделей представлены в таблице 6

В таблице 7 представлены результаты измерения Precision для каждой из используемых моделей.

В таблице 8 представлены результаты измерения Recall для каждой из используемых моделей.

В таблице 9 представлены результаты измерения F-меры для каждой из используемых моделей.

Модель	Логистическая регрессия	Метод опорных векторов	Дерево решений	Метод случайного леса
TP	11	11	12	8
TN	14	14	13	17
FP	4	3	3	3
FN	2	3	3	3

Таблица 6: Точности предсказательных моделей.

Модель	Логистическая регрессия	Метод опорных векторов	Дерево решений	Метод случайного леса
Precision	0.78	0.82	0.81	0.85

Таблица 7: Precision предсказательных моделей.

Модель	Логистическая регрессия	Метод опорных векторов	Дерево решений	Метод случайного леса
Recall	0.88	0.82	0.81	0.85

Таблица 8: Recall предсказательных моделей.

Модель	Логистическая регрессия	Метод опорных векторов	Дерево решений	Метод случайного леса
F-мера	0.82	0.82	0.81	0.85

Таблица 9: F-меры предсказательных моделей.

3.6 Критерий

Модели машинного обучения обычно принимают на вход вектор фиксированной размерности. Но количество измерений интервалов RR и QT отличается у разных пациентов, поэтому, чтобы обеспечить соответствие каждому пациенту только одного вектора признаков одинаковой размерности и улучшить точность ее предсказания, был разработан критерий, который отражает зависимость принадлежности пациента к определенной группе от интервалов RR, QT и ритма сердца пациента во время ЭКГ. Он вычисляется по формуле:

$$log_{\frac{RR}{QT}*Ritm}\frac{RR*QT*Ritm}{count},$$

где

RR – сумма интервалов RR,

QT – сумма интервалов QT,

$Ritm$ – вид ритма пациента,

$count$ – количество ЭКГ.

Показатели, которые использовались для вычисления критерия были выбраны исходя из их важностей.

Важности показателей

Важность показателя – это присвоение оценки каждому из входных параметров, которая показывает, насколько данная характеристика важна для предсказательной модели. Для вычисления этого показателя использовалась функция "feature_importances_"из библиотеки sklearn. Важность характеристики или ее еще называют важность Джини рассчитывается, как сумма количества разделений, которые используют данный признак, пропорционально количеству выборок. В таблице 10 приведены важности каждого из показателей.

Параметр	Важность
Возраст	0.18404
Пол	0.03784
Количество дней госпитализации	0.56535
Было ли изменение ритма	0.00000
Ритм	0.21277

Таблица 10: Таблица важностей показателей

На основании данной таблицы для вычисления критерия был выбран показатель, имеющий наибольшую важность и не зависящий от даты поступления человека в больницу – ритм сердца человека. Данную характеристику можно измерить не дожидаясь выписки или смерти человека.

Чтобы убедиться в его значимости, был проведен эксперимент, в котором для предсказания группы пациента были использованы данные с критерием и данные без критерия. Результаты сравнения приведены в таблице 11.

Таким образом, критерий позволил улучшить предсказательные модели.

Модель	Данные	Данные с критерием
Логистическая регрессия	0.6265	0.6564
Обучающее дерево	0.6467	0.6716
Метод опорных векторов	0.6161	0.6729
Метод случайного леса	0.6783	0.6845

Таблица 11:
Таблица сравнения точности моделей с критерием и без него.

3.7 Комбинирование моделей

Одним из методов работы с данными маленького размера является комбинирование моделей. Объединение прогнозов может способствовать улучшению точности предсказаний. Результат комбинирования моделей представлен в таблице 12.

Модели	Точность
RandomForest + LogReg + SVM	0.6606
RandomForest + LogReg	0.6603
LogReg + SVM	0.5941
RandomForest + SVM	0.6632

Таблица 12:
Таблица сравнения точностей комбинирования моделей

Результаты предсказания комбинирования моделей не улучшили точность предсказания, поэтому в программном комплексе использовалась модель с лучшими результатами – случайный лес.

Глава 4. Программная реализация

В данной главе описываются разработанные программные модули для введения данных о пациентах и визуализации результата предсказаний.

4.1 Описание используемых программных средств

В качестве языка программирования был выбран Python 3,7 [22], так как у него есть много фреймворков, которые значительно упрощают работу, а так же это один из самых популярных языков для задач машинного обучения [15]. Работа проводилась в среде разработки PyCharm [23]. Использовались такие библиотеки, как:

1. NumPy для работы с данными [3];
2. scikit-learn для применения методов машинного обучения [2];
3. Matplotlib для визуализации результатов [4].

В рамках данного проекта необходимо было создать приложение, которое будет использоваться врачами, поэтому у него должен быть удобный интерфейс. Так же плюсом приложения является и кроссплатформенность, так как это дает возможность запустить приложение на разных системах. Исходя из данных соображений рассматривались следующие фреймворки: PyQT [19], Tkinkter [20], Kivy [21].

Результаты анализа фреймворков приведены в таблице 13.

Для дальнейшей разработки проекта необходима новая библиотека, которая позволит развивать интерфейс в соответствии с современными требованиями. Вследствие чего после анализа преимуществ и недостатков вышеперечисленных инструментов был выбран PyQT, так как он отвечает всем поставленным требованиям и не имеет существенных недостатков.

Фреймворк	Преимущества	Недостатки
PyQT	1.Большой, мощный фреймворк 2. Много возможностей 3. Комбинация библиотеки QT и языка программирования Python	Необходимы обширные знания разработчика
Tkinter	1. Легкий в использовании 2. Большое интернет-сообщество	Очень старая библиотека, которая не может отвечать современным требованиям.
Kivy	Библиотека, предназначенная для разработки приложений, требующих инновационных пользовательских интерфейсов	Язык kV не подходит для компиляции кода. Следовательно, нужно будет смешивать языки ру и kV.

Таблица 13: Анализ фреймворков

4.2 Реализация

4.2.1 Программный комплекс

В рамках данного проекта был разработан программный комплекс, который позволяет вводить данные пациента. И на основе полученных данных показывает вероятность, с которой человек умрет. На рисунке 11 изображен пример работы приложения.

4.2.2 Комбинирование моделей

Для реализации комбинирования моделей был использован инструмент StackingClassifier библиотеки mlxtend. Стекинг - это метод ансамблевого обучения, позволяющий комбинировать несколько моделей классификации с помощью метаклассификатора. Модели классификации независимо обучаются на обучающей выборке, а затем мета-классификатор обучается по предсказанным классам моделей классификации.

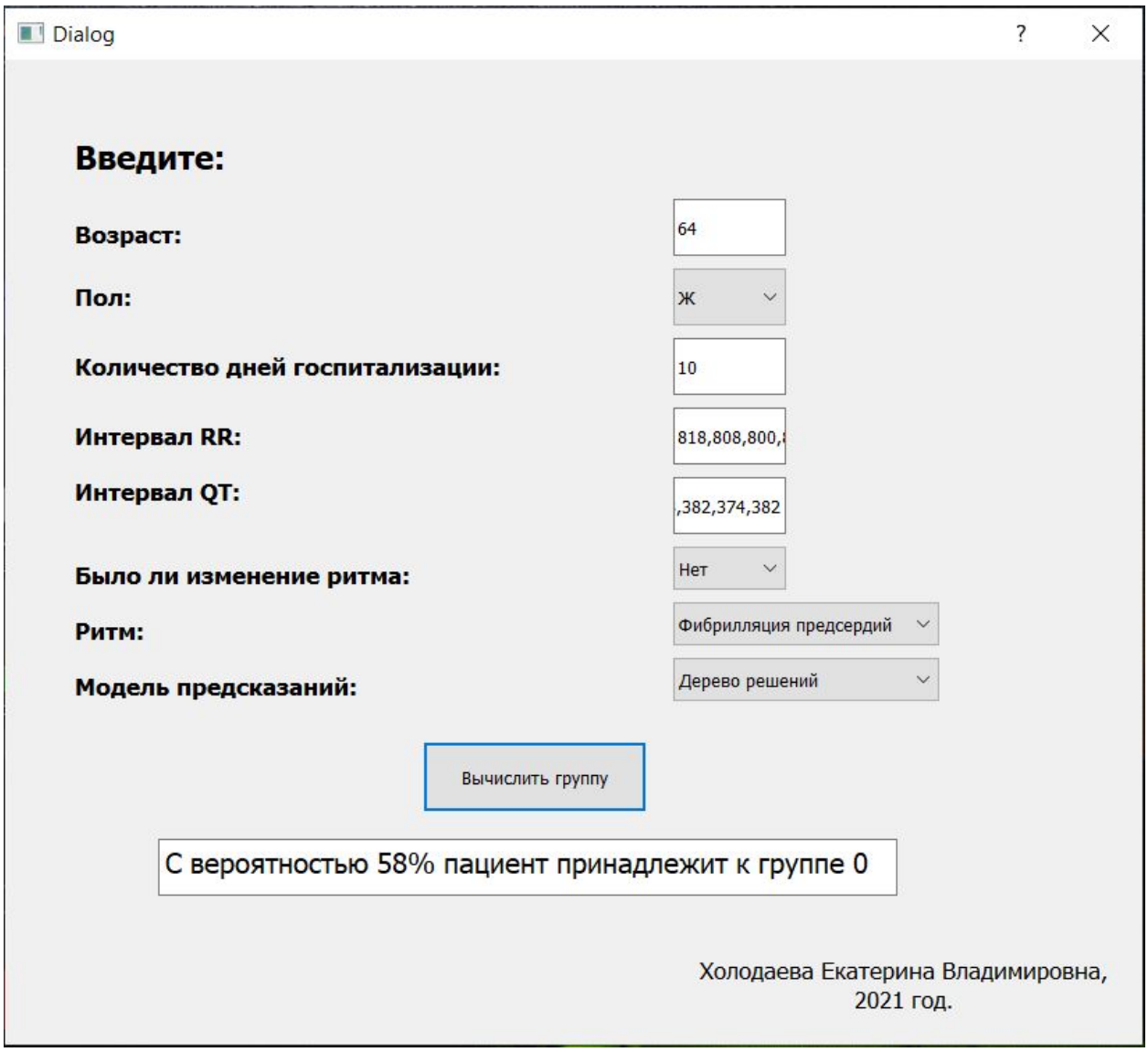

Рис. 11: Пример работы приложения

4.2.3 Кроссвалидация

Для реализации кроссвалидации использовался инструмент библиотеки scikit-learn GridSearchCV. Он не только подбирает лучшие параметры, но и позволяет производить деление тренировочной выборки более чем на 2 части, находя оптимальные параметры для каждого деления.

4.3 Архитектура приложения

На рисунке 12 изображена архитектура разработанного приложения.

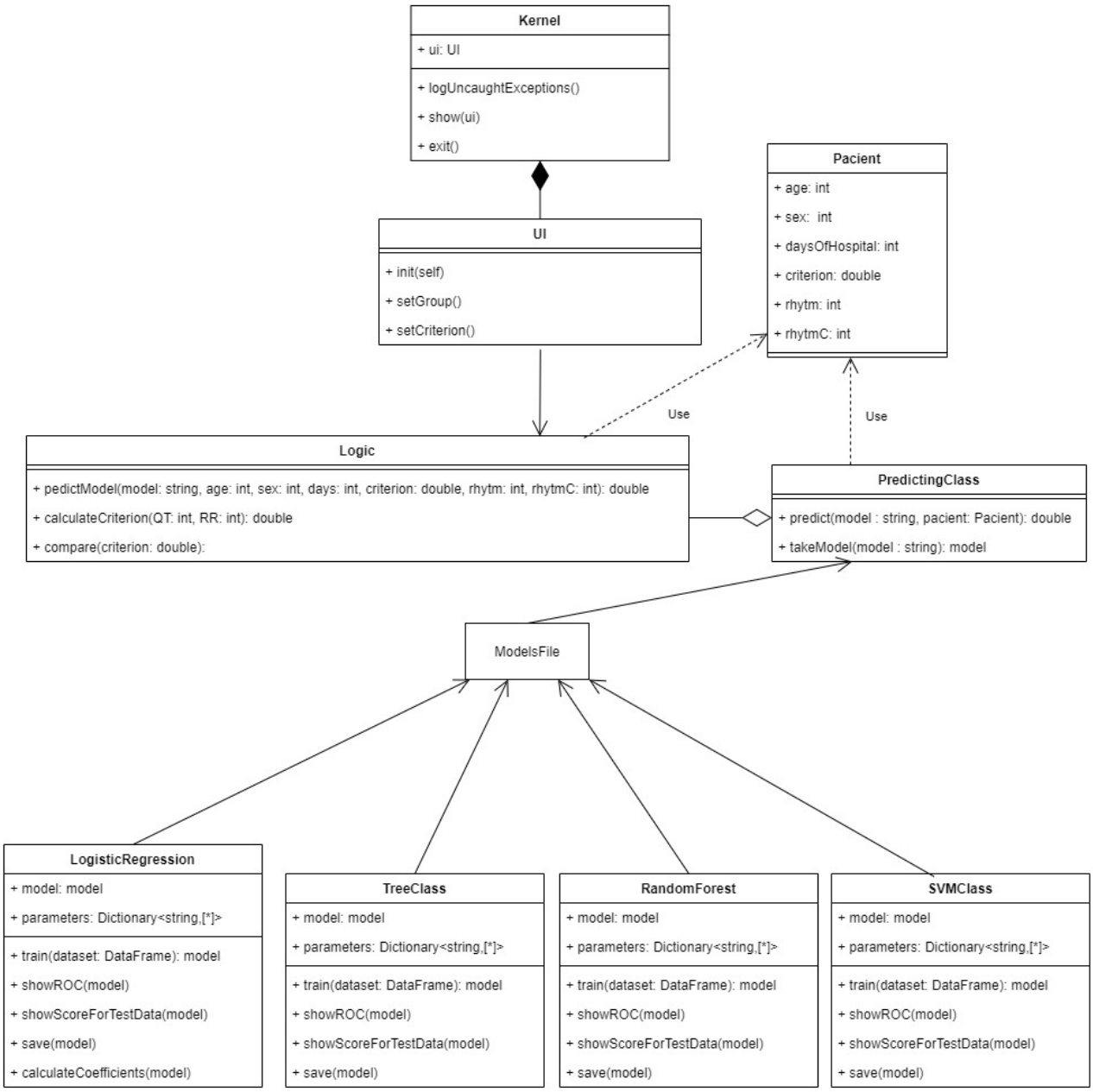

Рис. 12: Архитектура приложения

4.3.1 Kernel

Класс Kernel является главным классом, с которого начинается запуск приложения, метод show() инициализирует форму приложения, метод logUncaughtExceptions() отвечает за вывод информации о возникающих ошибках, метод exit() завершает работу приложения.

4.3.2 Компоненты методов

Для каждой из моделей был создан отдельный компонент.(Компоненты LogisticRegression, TreeClass, SvmClass, RandomForest). Обученная модель сохраняется в файл ModelsFile, чтобы не производить обучение каждый раз, когда пользователь будет вводить данные. Данные компоненты содержат методы: train() для обучения данных, showRoc() для визуализации ROC-кривой, showScoreForTestData() для визуализации результата на тестовых данных и save() для сохранения модели в файл.

4.3.3 PredictingClass

Данный класс содержит методы:

1. takeModel(), который извлекает модель, запрошенную пользователем, из файла. Принимает на вход строку, обозначающую метод предсказания. В качестве результата возвращает конкретную модель.

2. predict(), который используется для получения вероятности принадлежности пациента к группе 0 моделью, полученной на выходе метода takeModel().

4.3.4 Patient

Для реализации сущности пациента был создан класс Patient, который содержит все характеристики обследуемого человека.

4.3.5 Logic

Класс Logic содержит логику приложения, является связующим звеном между интерфейсом и другими классами. Он преобразует полученные характеристики в экземпляр класса Patient. Компонент содержит методы predictModel(), который принимает на вход данные, полученные от интерфейса и обрабатывает их для передачи классу PredictingClass(), полученные результаты передаются классу UI для отображения. Методы

calculateCriterion() и compare() являются вспомогательными для вычисления критерия пациента по интервалам RR и QT и сравнения с заранее заданной константой.

4.3.6 UI

Класс UI отвечает за графический интерфейс приложения. В данном классе происходит взаимодействие с QT Designer.

Заключение

Данная выпускная работа бакалавра была посвящена реализации методов машинного обучения для прогнозирования итога ухудшения состояния пациента с симптомами ССЗ, а также созданию приложения для визуализации результата. Что имеет большое значение для работы врачей, так как поможет им предотвратить смерти людей, уделив большее внимание пациентам, которым угрожает опасность. Полученные результаты формируют задел для дальнейшей работы по улучшению точности предсказательных моделей и нахождению новых статистически значимых критериев.

В ходе выполнения данной работы были получены следующие результаты:

1. исследованы существующие методы машинного обучения, проведен анализ релевантных работ с целью выбрать наиболее подходящие в рамках задачи модели;

2. реализовано несколько моделей предсказания, произведен их сравнительный анализ;

3. разработан критерий с целью использования показателей ЭКГ для обучения моделей, улучшающий их точность;

4. разработана архитектура прототипа программного модуля для визуализации оценки предсказания моделей и все полученные методы реализованы в данном прототипе.

Ссылка на репозиторий, где ведется разработка:
https://github.com/HolodaevaEkaterina/DiplomaProject

Список литературы

[1] Всемирная организация здравохранения, `https://www.who.int/cardiovascular_diseases/ru` (дата обращения 10.11.2020)

[2] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V.and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P.and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. andCournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E.// «Scikit-learn: Machine Learning in Python». //Journal of Machine Learning Research – 2011, pp. 2825–2830.

[3] Charles R. Harris and K. Jarrod Millman and St'efan J. van der Walt and Ralf Gommers and Pauli Virtanen and David Cournapeau and Eric Wieser and Julian Taylor and Sebastian Berg and Nathaniel J. Smith and Robert Kern and Mattihttps://ru.overleaf.com/project/5fac3e9c7bd15b0c37d94b23 Picus and Stephan Hoyer and Marten H. van Kerkwijk and Matthew Brett and Allan Haldane and Jaime Fern'andez del R'ıo and Mark Wiebe and Pearu Peterson and Pierre G'erard-Marchant and Kevin Sheppard and Tyler Reddy and Warren Weckesser and Hameer Abbasi and Christoph Gohlke and Travis E. Oliphant // «Array programming with NumPy».// Nature.– 2020. -№7825. pp. 357–362 //

[4] Hunter, John D// «Matplotlib: среда 2D-графики».//Computing in science engineering. –2007. -№3. pp. 90–95

[5] Zhu, F, Xu, D., Liu, Y., Lou, K., He, Z., Zhang, H., Sheng, Y., Yang, R., Li, X., Kong, X., Zhang, H.// «Machine learning for the diagnosis of pulmonary hypertension».// Kardiologiya. –2020. -№6. pp. 96–101

[6] Su, X., Xu, Y., Tan, Z., Wang, X., Yang, P., Su, Y., Jiang, Y., Qin, S., Shang, L.// «Prediction for cardiovascular diseases based on laboratory data: An analysis of random forest model».// Journal of Clinical Laboratory Analysis. –2020. -№9. e23421

[7] Priyanga, P., Pattankar, VV, Sridevi, S (// «A hybrid recurrent neural network-logistic chaos-based whale optimization framework for heart disease prediction with electronic health records».// COMPUTATIONAL INTELLIGENCE. –2021. -№1. pp.315-343.

[8] Siddiqui SY, Athar A, Khan MA, Abbas S, Saeed Y, Khan MF, Hussain M // «Modelling, Simulation and Optimization of Diagnosis Cardiovascular Disease Using Computational Intelligence Approaches».//JOURNAL OF MEDICAL IMAGING AND HEALTH INFORMATICS. –2019. -№5. pp.1005-1022

[9] Li, P., Hu, Y. Author, Liu, Z.-P.// «Prediction of cardiovascular diseases by integrating multi-modal features with machine learning methods».//Biomedical Signal Processing and Control. –2021. -№66. 102474

[10] Yang CX, Aranoff ND, Green P, Tavassolian N// «Classification of Aortic Stenosis Using Time-Frequency Features From Chest Cardio-Mechanical Signals».//TRANSACTIONS ON BIOMEDICAL ENGINEERING. –2020. -№6. pp. 1672-1683

[11] Tasnim, F., Habiba, S.U.// «A Comparative Study on Heart Disease Prediction Using Data Mining Techniques and Feature Selection».//ICREST 2021 - 2nd International Conference on Robotics, Electrical and Signal Processing Techniques. –2021. pp. 338-341

[12] Komal Kumar, N., Lakshmi Tulasi, R., Vigneswari, D.// «Performance Analysis of Classification Methods for Cardio Vascular Disease (CVD)».//Lecture Notes in Electrical Engineering. –2021. Volume 668. pp. 1231-1238

[13] Rajalakshmi, V., Sasikala, D., Kala, A// «A Predictive Analysis for Heart Disease Using Machine Learning».//Advances in Intelligent Systems and Computing. –2021. -№1172. pp. 473-479

[14] Zhijun Wu, Zhe Huang, Yuntao Wu, Yao Jin, Yanxiu Wang, Haiyan Zhao, Shuohua Chen, Shouling Wu, Xiang Gao// «Risk stratification for mortality

in cardiovascular disease survivors: A survival conditional inference tree analysis.».//Nutr Metab Cardiovasc Dis. –2021. pp. 420-428

[15] Matthieu Brucher, Matthieu Perrot, Edouard Duchesnay, David Cournapeau, Alexandre Passos, Jake Vanderplas, Vincent Dubourg, Ron Weiss, Peter Prettenhofer, Mathieu Blondel, Olivier Grisel, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion // «Scikit-learn: Machine Learning in Python.».//Journal of Machine Learning Research. –2011. pp. 2825-2830

[16] Max Kuhn, Kjell Johnson.// «Applied Predictive Modeling». –2013. doi:10.1007/978-1-4614-6849-3

[17] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani.// «An Introduction to Statistical Learning». –2015. doi:10.1007/978-1-4614-7138-7

[18] Федеральное государственное бюджетное учреждение «Всероссийский центр экстренной и радиационной медицины им. А.М.Никифорова» МЧС России [Электронный ресурс]
`https://nrcerm.ru/` (дата обращения 10.11.2020)

[19] PyQT [Электронный ресурс]
`https://www.qt.io/` (дата обращения 10.11.2020)

[20] Tkinkter [Электронный ресурс]
`https://tkdocs.com/` (дата обращения 10.11.2020)

[21] Kivy [Электронный ресурс]
`https://kivy.org/` (дата обращения 10.11.2020)

[22] Python [Электронный ресурс]
`https://www.python.org/` (дата обращения 10.11.2020)

[23] PyCharm [Электронный ресурс]
`https://www.jetbrains.com/ru-ru/pycharm/` (дата обращения 10.11.2020)

[24] Всемирная организация здравоохранения, статистика о ведущих причинах смертности [Электронный ресурс]
https://www.who.int/ru/news/item/09-12-2020-who-reveals-leading-causes-of-death-and-disability-worldwide-2000-2019 (дата обращения 10.11.2020)

[25] What to do with "small" data? [Электронный ресурс]
https://medium.com/rants-on-machine-learning/what-to-do-with-small-data-d253254d1a89 (дата обращения 10.11.2020)

[26] Руководство по измерению QT при проведении ЭКГ мониторинга в рамках внедрения новых лекарственных препаратов и краткосрочных схем лечения лекарственно-устойчивого туберкулёза [Электронный ресурс]
https://www.challengetb.org/publications/tools/pmdt/Guidance_on_ECG_monitoring_in_NDR_RUS.pdf (дата обращения 10.11.2020)

[27] Как выбрать метрики для валидации результата Machine Learning [Электронный ресурс]
http://blog.datalytica.ru/2018/05/blog-post.html?m=1 (дата обращения 10.11.2020)

[28] Кросс-валидация (Cross-validation) [Электронный ресурс]
https://long-short.pro/post/kross-validatsiya-cross-validation-304 (дата обращения 10.11.2020)